УДК 004.624

Е.И. Чумаченко, С.С. Захаров
НТУ Украины «Киевский политехнический институт», г. Киев
Украина, 03056, г. Киев, ул. Политехническая 41(18 корпус)

Алгоритмическое обеспечение распределенных баз данных

E.I. Chumachenko, S. S. Zakharov
NTU of Ukraine «Kiev Polytechnical Institute», Kiev
Ukraine, 03056, Kiev, Politekhnicheskaya St. 41 (18th case)

Algorithmic Providing the Distributed Databases

О.І. Чумаченко, С.С. Захаров
НТУ України «Київський політехнічний інститут», м. Київ
Україна, 03056, м. Київ, вул. Політехнічна 41(18 корпус)

Алгоритмічне забезпечення розподілених баз даних

В данной статье рассмотрены основные два алгоритма, используемые в построении распределённых баз данных. Описана структура и принципы работы генетического и эвристического алгоритма и приведены наглядные примеры воплощения их в процессе создания РБД. Внедрение в работу РБД подобных алгоритмов значительно автоматизируют и повышают эффективность работы каждого процесса и всей базе данных в целом.
Ключевые слова: распределённые базы данных, базы данных, алгоритмы, программирование, генетический алгоритм, эвристический алгоритм.

Two main algorithms that are used in distributed databases constructing have been observed in this article. Genetic and heuristic algorithms operating structure and principles were described and their embodiment in the course of DDB creation were shown on clear examples. Such algorithms implementation into DDB operating considerably automates and promotes efficiency of both any process and database in whole.
Keywords: distributed databases, databases, algorithms, programming, genetic algorithm, heuristic algorithm.

В даній статті розглянуто основні два алгоритми, використані в побудові розподілених баз даних. Описана структура і принципи роботи генетичного і евристичного алгоритму і наведені наочні приклади втілення їх в процесі створення РБД. Впровадження в роботу РБД подібних алгоритмів значно автоматизують і підвищують ефективність роботи кожного процесу і всій бази даних в цілому.
Ключові слова: розподілені бази даних, бази даних, алгоритми, програмування, генетичний алгоритм, евристичний алгоритм.

Введение

Создание распределенных информационных систем является весьма актуальной задачей. Это связано с возрастающими потребностями в приложениях. Увеличиваются требования к оперативности и своевременности информации. Управление информацией происходит с помощью систем управления базами данных (СУБД). Для достижения высокой производительности распределенных приложений, работающих с базами данных, необходимы эффективные методы проектирования распределенных баз данных (РБД).

В статье рассмотрен кратко цикл проектирования РБД, показана сложность решаемой задачи и применимость генетических алгоритмов для решения поставленной задачи.

Цель работы. Целью работы является разработка методов проектирования распределенной базы данных, которые описывают способ разбиения централизованной БД на фрагменты и размещения полученных фрагментов в узлах заданной вычислительной сети (ВС). На основании полученных методов необходимо создать алгоритмы автоматизированной системы, результатами работы которой, помимо схем фрагментации и размещения БД, будут рекомендации по повышению эффективности обработки запросов к РБД.

Проектирования РБД

Состоит из двух фаз: начальное проектирование и репроектирование. В большинстве случаев под начальным проектированием понимают фрагментацию БД и размещение фрагментов по узлам вычислительной сети (ВС). С течением времени возможно ухудшение производительности приложений, работающих с РБД, вызванное изменениями в инфраструктуре распределенной среды (ИРС). Под инфраструктурой понимаются физические и логические параметры функционирования системы, а именно транзакции и их частоты, топология ВС, характеристики узлов ВС и т.д.

Изменения ИРС могут быть двух типов: физические и логические. Возможны три сценария репроектирования РБД: корректирующее, превентивное и адаптивное.

Сложность решаемых задач. Задача проектирования РБД формулируется так: для данной логической схемы БД, множества запросов и конфигурации ВС описать схему фрагментации, схему размещения фрагментов и стратегии исполнения каждого запроса таким образом, чтобы оптимизировать целевую функцию.

В [1], [2] показано, что оптимальные алгоритмы фрагментации и размещения данных являются NP-трудными, т.е. с ростом размерности задач их вычислительная сложность растет экспоненциально.

При формировании стратегии исполнения реляционного запроса необходимо принять решения о способе и порядке выполнения операций соединения (join) отношений, а также об используемых методах параллельной обработки. Рассматриваемые решения приведены в табл. 1.

Таблица 1 – Количество вариантов исполнения запроса

Решение	Количество вариантов
Выбор фрагмента	Πri, где ri – количество
Последовательность операции	$(t-1)!$, где t – количество
Узел, в котором производится операция	
Применение операции semi-join	$3(t-1)$
Применение double-pipelined hash join	$3(t-1)$
Степень внутриоператорного	$t(n)!$

Общее количество возможных стратегий исполнения запроса равно произведению количества вариантов каждого решения. Например, для сети из 4-х узлов и запроса, в котором используются 5 фрагментов, каждый из которых имеет 2 реплики, существует почти 16×10^6 возможных вариантов.

Эвристические алгоритмы. Для решения NP-полных задач применяются эвристические алгоритмы, которые уменьшают вычислительную сложность и позволяют получить близкое к оптимальному решению. Существуют следующие типы эвристических моделей:

– модель слепого поиска, которая опирается на так называемый метод проб и ошибок;

– лабиринтная модель, в которой решаемая задача рассматривается как лабиринт, а процесс поиска решения – как блуждание по лабиринту;

– структурно-семантическая модель, которая считается в настоящее время наиболее содержательной. Она отражает семантические отношения между объектами, составляющими область задачи.

В [3] отмечено, что для решения задач кластеризации и компоновки наиболее успешно применяются генетические алгоритмы (ГА). В ГА любое решение задачи синтеза представляется хромосомой, состоящей из генов. Значениями генов являются значения проектных параметров. Направленный перебор решений осуществляется с помощью генетических операторов выбора родителей, кроссовера, мутации, селекции, переупорядочения.

Сначала формируется исходное поколение, состоящее из g хромосом. Размер поколения выбирается таким образом, чтобы в достаточной степени был представлен набор существующих решений. Например, поколение из 100 хромосом обычно достаточно для того, чтобы осуществить поиск среди 10^9 доступных решений. Хромосомы оцениваются с использованием функции приспособленности.

Далее случайным образом среди хромосом данного поколения выбираются пары родителей, причем вероятность выбора хромосом с лучшими значениями функции приспособленности должна быть выше. Следующее поколение образуется (селектируется) из g перспективных дочерних хромосом, являющихся результатом ряда операций кроссовера. Кроссовер заключается в разрыве двух родительских хромосом и рекомбинировании образовавшихся хромосомных отрезков. Мутации, т.е. случайные изменения генов, происходят с заданной вероятностью и служат для исключения застревания поиска в ограниченном подпространстве. Разновидности генетических операторов и их сочетаний порождают множество ГА, описание которых можно найти в [4].

Применение генетических алгоритмов. Задача проектирования РБД формируется тремя NP-полными взаимозависимыми задачами, т.е. входными данными для следующей задачи является решение предыдущей. Таким образом, в методе получения проекта РБД, близкого к оптимальному, целесообразно применять вложенные генетические алгоритмы. Подходящий алгоритм для такой задачи выглядит следующим образом [5].

Для каждого отношения R на основании запросов определяется набор минимальных фрагментов. Каждый минимальный фрагмент характеризуется множеством минтерм предикатов и группой атрибутов.

После этого формируется хромосома, описывающая размещение фрагментов на узлах ВС. Если фрагмент размещен в узле, то в соответствующую ячейку ставится 1, в противном случае 0. Структура хромосомы приведена в табл. 2.

Таблица 2 – Структура хромосомы схем фрагментации и размещения

Узлы\	Отношение 1				Отношение 2			
	Атрибуты		Предикаты		Атрибуты		Предикаты	
фрагменты	Атр1	Атр2	Пред1	Пред2	Атр1	Атр2	Пред1	Пред2
Узел1	1	0	1	1	0	1	0	1
Узел2	0	1	1	0	1	1	1	0

Для удовлетворения свойства полноты в каждой колонке должна быть хотя бы одна единица. Каждому атрибуту и предикату назначено значение мощности: для атрибутов мощность определена на основании типа данных, для предикатов – селективности. На основании мощности определяется количество возвращаемых запросом данных.

Для каждой хромосомы из сформированного поколения схем фрагментации и размещения запускается вложенный генетический алгоритм, направленный на получение оптимальной стратегии исполнения запросов. Хромосомы вложенного алгоритма сформированы из решений, принимаемых при исполнении запроса. Функцией приспособленности вложенного алгоритма является критерий оптимальности РБД. После определенного количества поколений значение функции приспособленности лучшей хромосомы из вложенного алгоритма используется как оценка функции приспособленности внешнего алгоритма. Комбинация хромосом из внешнего и вложенного алгоритмов полностью описывают проект РБД.

В разрабатываемом программном обеспечении генетический алгоритм реализован следующим образом (представлен главный цикл Генетического алгоритма реализуемый в программе).

```
…………

// инициализация
        int[] solutions = new int[n];
        for (int i = 0; i < n; i++) solutions[i] = rnd.Next(32);
        Array.Sort(solutions);

        // главный цикл

        while (solutions [n - 1] < 31)
           {
           // отбор

           int parent 1 = solutions [rnd.Next(n)], parent2 = solutions[rnd.Next(n)];

// рекомбинация

           int mask = (1 << rnd.Next(5)) - 1;

           int child = parent 1 & mask | parent2 & ~mask;

// мутация потомка
           child ^= 1 << rnd.Next(l0) & 31;
           if (child > solutions[0]) // потомок лучше самого плохого?

           {

              solutions[0] = child; // меняем самого плохого

              Array.Sort(solutions);
        }
        }
        …..................
```

Заключение

У людей, пользовавшихся генетическими алгоритмами, часто создается впечатление, что они неэффективны. Но, на наш взгляд, виновата не генетическая идея, а способ ее реализации. Генетическая модель представляет собой множество особей. Обычно особь – это кортеж значений переменных задачи. Количество информации, которое может хранить такая модель – не более чем двоичный логарифм от количества особей (это при отсутствии клонов в популяции). А количество информации в задаче – это двоичный логарифм всех возможных комбинаций значений переменных. Накопление информации позволяет экспоненциально увеличить эффективность алгоритмов, так как экспоненциально снижается количество возможных вариантов в «пространстве поиска». А генетический алгоритм этого не делает, поэтому он в общем случае и не эффективен.

Основной перспективой исследования распределённых баз данных является:

1. Отход от ограничений в количестве хранимой информации.

2. Более надёжная система хранения данных, так как информация находится на нескольких сетевых серверах.

3. Гибкая система работы с другими программами.

В статье рассмотрен жизненный цикл проектирования РБД и описаны его этапы. Сформулирована задача проектирования РБД и оценена ее сложность. Обоснован выбор генетических алгоритмов для решения поставленной задачи и предложен подход, позволяющий учесть взаимозависимость этапов проектирования.

Литература

1. Evolutionary Algorithms for Allocating Data in Distributed Database Systems / Ahmad I., Karlapalem K., Kwok Y.K., et al. // Distributed and Parallel Databases. - 2002. - Vol. 11, № 1. – P. 5-32.
2. Apers P.M.G. Data allocation in distributed database systems / P.M.G. Apers // ACM Transactions on Database Systems. – 1988. – Vol. 13, № 3. – P. 263-304.
3. Норенков И.П. Эвристики и их комбинации в генетических методах дискретной оптимизации / И.П. Норенков // Информационные технологии. – 1999. – № 1. – С. 2-7.
4. Батищев Д.И. Генетические алгоритмы решения экстремальных задач : уч. пособие / Д.И. Батищев. – Воронеж : ВГТУ, 1995. – 69 с.
5. Новосельский В.Б. Методы автоматизации проектирования распределенных баз данных / В.Б. Новосельский. – СПб. : СПбГУ ИТМО, 2008.

References

1. Ahmad I. Distributed and Parallel Databases. 2002. Vol. 11. № 1. P. 5-32.
2. Apers P.M.G. ACM Transactions on Database Systems. 1988. Vol. 13. №3. P. 263-304.
3. Norenkov I.P. Informacionnye tehnologii. 1999. №1. S. 2-7.
4. Batishhev D.I. Geneticheskie algoritmy reshenija jekstremal'nyh zadach. Ucheb. posobie. Voronezh: VGTU. 1995. 69 s.
5. Novosel'skij V. B. Metody avtomatizacii proektirovanijaraspredelennyh baz dannyh. SPb.: SPbGU ITMO. 2008.

RESUME

E.I. Chumachenko, S. S. Zakharov

Distributed Databases Algorithmic Providing

Currently almost all of organizations use databases (DB) technologies. Increasing complexity of means measuring, computing and presenting the information they have increasing demands on more and more information control as a result. It leads DB sizes to exceed centralized systems physical limits, thus, decentralization processes and creating distributed databases become more important. To get high productivity of distributed applications operating with distributed databases they need effective algorithms and DDB projecting methods.

In this article two basic algorithms solving the task are presented. Genetic algorithm that helps us to solve the task is described in details. It's also shown the block from program code where directly genetic algorithm is realized visually.

Статья поступила в редакцию 01.10.2012.